# **Technical specifications for software** «Development of a system for Fibonacci Number Calculator »

**Done By Rami Alshalabi   
SE (eng) 1-21**

## Fibonacci Number Calculator: Requirements Specification

**1. Introduction**

This document outlines the requirements for a Fibonacci Number Calculator program. The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones, usually starting with 0 and 1. This calculator will allow users to find Fibonacci numbers based on their position in the sequence.

**2. Goal**

The primary goal of this program is to create a reliable and user-friendly tool for calculating Fibonacci numbers based on their position in the sequence. The program should provide accurate results for a reasonable range of inputs while handling edge cases appropriately.

**3. Functional Requirements**

**3.1 Fibonacci Calculation**

* The program shall calculate the Fibonacci number for a given position in the sequence.
* The calculation shall be accurate for all non-negative integers up to and including the 78th position.

**3.2 Input Handling**

* The program shall accept user input for the position of the desired Fibonacci number.
* It shall validate that the input is a non-negative integer.

**3.3 Error Handling**

The program shall provide clear error messages for invalid inputs, including:

* Non-number inputs
* Negative numbers
* Non-integer numbers
* Numbers greater than 78 (due to potential integer overflow)

**3.4 User Interface**

* The program shall provide a graphical user interface for input and output.
* This interface shall include:
  + An input field for the user to enter the desired position.
  + A button to trigger the calculation.
  + An area to display the calculated Fibonacci number or an appropriate error message.

**3.5 Result Display**

* The program shall display the calculated Fibonacci number clearly to the user.
* For valid inputs, the program shall show both the input position and the corresponding Fibonacci number.

**4. Non-Functional Requirements**

**4.1 Performance**

* The program shall calculate and display results for valid inputs (0-78) within 100 milliseconds on average hardware.

**4.2 Accuracy**

* The program shall provide 100% accurate results for all valid inputs (0-78).

**4.3 Usability**

* The user interface shall be intuitive and easy to use, requiring no special training.
* Error messages shall be clear and understandable to users without a technical background.

**4.4 Compatibility**

* The program shall be compatible with modern web browsers including Chrome, Firefox, Safari, and Edge.

**4.5 Responsiveness**

* The user interface shall be responsive and adapt to different screen sizes, from mobile devices to desktop computers.

**4.6 Accessibility**

* The program shall be accessible to users with disabilities, following WCAG 2.1 Level AA guidelines.

**4.7 Maintainability**

* The code shall be well-documented and follow best practices for readability and maintainability.
* The program structure shall allow for easy updates and potential future enhancements.

**4.8 Scalability**

* While the current implementation is limited to the 78th Fibonacci number, the program structure should allow for potential future expansion to handle larger numbers.

**4.9 Security**

* The program shall validate and sanitize all user inputs to prevent injection attacks or other security vulnerabilities.

**4.10 Testability**

* The program shall include a comprehensive set of unit tests covering both normal operations and edge cases.

**5. Conclusion**

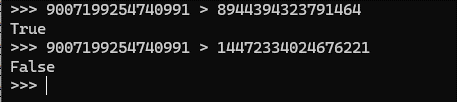
This requirements specification provides a comprehensive guide for the development of the Fibonacci Number Calculator. It outlines the essential functionality, user experience considerations, and important non-functional aspects such as performance, accuracy, and maintainability. By adhering to these requirements, the resulting program will be a robust, user-friendly tool for calculating Fibonacci numbers, built to high standards of software engineering.

# Test Cases:

| **Test** | **Serial number** | **Expected result** | **Actual result** |
| --- | --- | --- | --- |
| **1** | **0** | **0** | **0** |
| **2** | **1** | **1** | **1** |
| **3** | **5** | **5** | **5** |
| **4** | **10** | **55** | **55** |
| **5** | **20** | **6765** | **6765** |
| **6** | **78** | **8944394323791464** | **8944394323791464** |
| **7** | **-1** | **Error** | **Error: Input must be a non-negative number** |
| **8** | **3.5** | **Error** | **Error: Input must be an integer** |
| **9** | **'abc'** | **Error** | **Error: Input must be a number** |
| **10** | **79** | **Error** | **Error: Input is too large, may cause integer overflow** |

1. Tests:

Why we used the limit of 78?  
because in js the limit of an integer is:



List of Fibonacci numbers to compare the answers with:   
https://planetmath.org/listoffibonaccinumbers